UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/599,170	09/21/2006	Ryuta Nakanishi	P30744	5285

52123 7590 05/28/2009
GREENBLUM & BERNSTEIN, P.L.C.
1950 ROLAND CLARKE PLACE
RESTON, VA 20191

EXAMINER	
DUDEK JR, EDWARD J	

ART UNIT	PAPER NUMBER
2186	

NOTIFICATION DATE	DELIVERY MODE
05/28/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

gbpatent@gbpatent.com
pto@gbpatent.com

Office Action Summary	**Application No.** 10/599,170	**Applicant(s)** NAKANISHI ET AL.	
	Examiner Edward J. Dudek	**Art Unit** 2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 April 2009*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-17 and 19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1, 3-17, and 19* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *16 April 2009* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.☒ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *12-11-08*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

This Office Action is responsive to the amendment filed on 16 April 2009 in application #10/599,170.

Claims 1, 3-17, and 19 are pending and have been presented for examination.

Claims 2 and 18 have been cancelled.

Response to Arguments

Applicant's arguments, see page 11, filed 16 April 2009, with respect to the drawings have been fully considered and are persuasive. The objection to the drawings has been withdrawn.

Applicant's arguments, see page 11, filed 16 April 2009, with respect to claim 16 have been fully considered and are persuasive. The rejection of claim 16 under 35 U.S.C. § 112, second paragraph, has been withdrawn.

Applicant's arguments, see page 15, filed 16 April 2009, with respect to claims 1, 3-4, 6, and 18 have been fully considered and are persuasive. The rejection of claims 1, 3-4, 6, and 18 under 35 U.S.C. § 102(b) as being anticipated by Shinozaki have been withdrawn.

Applicant's arguments filed 16 April 2009 with respect to the rejection of claims 1, 3-11, and 18 under 35 U.S.C. § 102(b) as being anticipated by Afsar have been fully considered but they are not persuasive.

Applicant argues "At most, the asserted portion of Afsar may be considered to teach that the next data prefetch controller selection process involves examining an instruction line, preferably related to an executed instruction, for a particular characteristic. However, Afsar does not teach or suggest that the information for generating the new condition is obtained from a process, as specified in Applicants' independent claim 1." (See page 12)

The Examiner respectfully disagrees. Afsar utilizes a data prefetch engine which generates a prefetch prediction table. The table is generated using information relating to the present instruction that is being executed on the processor (see column 6, lines 38-56). This information is obtained from the processor (see column 5, lines 1-18). The program counter and addressing modes are stored in registers in the processor, and this information is transferred from the registers to the data prefetch engine (see figures 4 and 7). Furthermore, the data prefetch engine is part of the processor, so all the information that is obtained and generated is within the processor (see column 3, lines 50-60).

Applicant argues "Applicant independent claim 1 recites, *inter alia*, a judger that judges, using the condition held by said condition holder, whether a current state of the processor satisfies the condition." Applicant further argues "The Examiner asserts column 8, lines 1-17 of Afsar as teaching the above-noted features of Applicants' independent claim 1. It is submitted that the asserted portion of Afsar merely discloses generating a data prefetch prediction table. Further, Afsar is submitted to disclose determining whether a current program counter from a processor matches a program counter tag of a data prefetch prediction table. Afsar does not disclose that a current state of the processor satisfies the condition, as recited in independent claim 1." (See page 13)

The Examiner respectfully disagrees. The cited section does not disclose the generation of a data prefetch table, the cited section teaches how the data prefetch table is utilized after it is generated. The program counter from the process is transferred from the processor to the data prefetch predictor (which is part of the data prefetch engine) and the prefetch predictor determines if the program counter that is received matches a program counter tag in the table. The prefetch predictor is the judger, since this unit judges whether the program counter matches a condition held by the table, condition holder). The program counter is representative of a current state of a processor. At the time the program counter is transferred to the prefetch predictor, that is the current state of the processor. At that time the program counter held the value that is sent to the prefetch predictor. The program counter was not altered or manipulated in any way.

Applicant argues "Further, Applicants' independent claim 1 recites, *inter alia*, a manipulator that manipulates a cache according to the manipulation command held by the command holder using the address generated by said address generator when the judger judges that the condition is satisfied and an address generator generates an address to be manipulated. The asserted portion of Afsar in column 8, lines 18-64 merely discloses obtaining prefetch data from a main memory and transferring the data to a cache, when a prefetch miss occurs and a refill manager is called upon to locate the prefetch data. That is, a processor constructed according to the teaching of Afsar cannot and does not provide a manipulation command to the cache memory though an instruction." (See page 13)

The Examiner respectfully disagrees. The refill manager is part of the data prefetch engine (see figure 4, elements 56 and 84), and the data prefetch engine is part of the processor (See column 3, lines 50-60). The commands that are sent from the refill manager at essentially being sent from the processor since the refill manager is contained within the processor. When the refill manager attempts to locate the needed data at the cache, it is inherent that an instruction is sent to fetch the data. If the data is not there, then another instruction is sent to the main memory to obtain the data and then store the data in the cache. By storing this new data in the cache, the cache memory is being manipulated.

Applicant argues "However, Afsar does not teach or suggest a new condition, let alone generating a condition using the current state of the process or the condition held by the condition holder and the information for generating the new condition held by said condition generation information holder, as recited in Applicants' independent claim 1.

The Examiner respectfully disagrees. Each time Afsar adds data to the prefetch prediction table, a new condition is being generated. Each entry in the table contains a unique program counter tag, and therefore, each time a new entry is added, there is now a new condition. The current state of the processor is used to generate these conditions. The table is generated using information relating to the present instruction that is being executed on the processor (see column 6, lines 38-56). This information is obtained from the processor (see column 5, lines 1-18). The program counter and addressing modes are stored in registers in the processor, and this information is transferred from the registers to the data prefetch engine (see figures 4 and 7). Furthermore, the data prefetch engine is part of the processor, so all the information that is obtained and generated is within the processor (see column 3, lines 50-60).

The Rejections are maintained as repeated below.

Claim Objections

Claims 12-13 and 15 are objected to because of the following informalities:

Claim 12 contains a typo at lines 10 and 12. It is suggested the word “transferer” be amended to “transferor”, and the word “registerer” be amended to read “registrar”.

Claim 13 contains a typo at line 10. It is suggested the word “registerer” be amended to read “registrar”.

Claim 15 contains the phase “... wherein said manipulator manipulates includes...” in line 2. It is suggested this phrase be amended to read "... wherein said manipulator includes...”, as this in consistent with other similar limitations in the other dependent claims.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims1, 3-17, and 19 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1 and 19 recite the limitation “a condition generator that generates a new condition using one of the current stat of the processor and the condition held by said condition holder, and the information that generates the new condition held by said condition generation information holder” in lines 15-17 and lines 12-15 respectively. It is not clear which limitations are used to generate the new condition. There are three limitations listed: "the current state of the processor", "the condition held by said condition holder", and "the information that generates the new condition held by said condition generation information holder". It is not clear if each limitation can be used on its own for generating the new condition, or if the first two limitations are to be used together and the third limitation used on its own because there is a comma separating the second and third conditions. Further more, in the response to arguments, see page 14, the first and second limitations are separated by an "or". Therefore, it is confusing how these three limitations are applied when generating a new condition. For purposes of examination, this will be interpreted as any one of the three limitations being able to be used to generate the new condition.

Claims 3-17 are also deficient as they depend from claim 1.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1, 3-11, and 19 are rejected under 35 U.S.C. 102(b) as being anticipated by Afsar (**U.S. Patent #6,401,193**).

As per claims 1 and 19: Afsar discloses a cache memory system comprising: a condition generation information holder that holds information that generates a new condition, the information that generates the new condition being obtained from a processor **(see column 6, lines 1-21)**; a command holder that holds a manipulation command given by the processor **(see column 8, lines 18-64)**; a condition holder that holds a condition relating to a state of the processor **(see column 8, lines 1-17)**; a judger that judges, using the condition held by said condition holder whether a current state of the processor satisfies the condition **(see column 8, lines 1-17)**; an address generator that generates an address to be manipulated **(see column 6, lines 37-55)**; a manipulator that manipulates a cache according to the manipulation command held by said command holder using the address generated by said address generator, when said judger judges that the condition is satisfied **(see column 8, lines 18-64)**; and a condition generator that generates a new condition using one of the current stat of the processor and the condition held by said condition holder, and the information that generates the new condition held by said condition generation information holder **(see**

column 6, lines 1-21, each time a new entry is added to the table a new condition is generated using the current state of the processor).

As per claim 3: wherein said condition generator generates a condition concerning a value of a specific register, within the processor **(see column 8, lines 1-16)**.

As per claim 4: wherein the specific register comprises a program counter **(see column 8, lines 1-16)**.

As per claim 5: wherein said condition generator generates, as the condition, one of a memory access within a specific address range and a memory access outside of the specific address range **(see column 6, lines 1-21)**.

As per claim 6: wherein said condition generator generates, as the condition, execution of a specified instruction by the processor **(see column 6, lines 1-21)**.

As per claim 7: wherein said condition generator generates the new condition by performing a specified calculation on a current condition **(see column 6, lines 22-36)**.

As per claim 8: wherein said condition generator generates a memory access address as the condition **(see column 6, lines 1-21)**; and wherein said condition generator generates the new condition by adding a constant to the current condition in a case where said judger judges that the condition is satisfied **(see column 6, lines 37-55)**.

As per claim 9: wherein the constant is one of: an increment value or a decrement value in a post-increment load/store instruction executed by the processor;

and a difference value of addresses in two load/store instructions executed by the processor **(see column 6, lines 37-55)**.

As per claim 10: wherein said condition generator generates plural conditions **(see column 7, lines 49-67)**, and wherein said judger judges whether all of the plural conditions are satisfied **(see column 8, lines 1-17)**.

As per claim 11: wherein said condition generator generates plural conditions **(see column 7, lines 49-67)**, and wherein said judger judges whether any of the plural conditions are satisfied **(see column 8, lines 1-17)**.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 12, and 14-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Afsar **(U.S. Patent #6,401,193)** in view of well known practices in the art.

As per claim 12: Afsar discloses all the limitations of claim 1 as discussed above. Afsar further discloses wherein said manipulator includes: a data judger that judges, in a case where said judger judges that the condition is satisfied, whether data corresponding to the address generated by said address generator is stored in the cache **(see column 8, lines 18-33)** and a transferor that transfers, from a memory to the selected line, the data corresponding to the address **(see column 8, lines 34-64)**.

Afsar fails to disclose a selector that selects a line within the cache in a case where it is judged that the data is not stored; an updater that performs a write back from the selected line when the selected line is valid and dirty; and a registerer that registers the address as a tag to the selected line. The system of Afsar involves pre-fetching data to a cache memory so the data is already present when the processor needs the data. Cache memories are relatively small and therefore can only hold a few entries. The data in the cache is cycled in and out throughout the execution of a program. As entries are removed from the cache, the data in the cache line must be written back to main memory if the data has been updated, this is what is known as a write back cache, and Official Notice is hereby taken. Furthermore, when a system is determining if data is present in a cache memory, the tag memory is searched. The tag memory contains addresses to the data that is stored in the cache. When a cache line is updated, the tag associated with the line is also updated, and Official Notice is hereby taken. It would have been obvious to a person having ordinary skill in the art to which said subject matter pertains to have selected a line for replacement in the cache memory, since space is limited and lines are replaced from time to time, perform a write back of data in the selected line if the data has been updated so that the updated data is not lost, and to register the address of the new data in the tag memory so the data line can be searched when looking for the data in the cache memory.

As per claim 14: Afsar discloses all the limitations of claim 1 as discussed above. Afsar further discloses wherein said manipulator includes: a data judger that judges, in a case where said judger judges that the condition is satisfied, whether data corresponding to the address generated by said address generator is stored in the cache **(see column 8, lines 18-33)** and a selector that selects a line, within the cache, in which data is stored in a case where it is judged that the data is stored **(see column 8, lines 18-30)**. Afsar fails to disclose an updater that performs a write back from the selected line when the selected line is valid and dirty. There are two methods for ensuring that data that is written to a cache memory is updated in main memory. A write though method which involves updating data in main memory whenever the data in the cache line is updated and a write-back method that involves updating data in the main memory whenever the data is evicted from the cache memory. Both techniques have their advantages. A write thought method puts more traffic on the memory bus as data is constantly being sent to the main memory to update any data lines that have been written to, but allows faster eviction as the data can just be cast out without having to first wait for the data in main memory to be updated. A write back technique reduces the amount of traffic on the memory bus, but requires a wait time when evicting the data since the data in the main memory must first be updated so that any changes made are not lost. Official Notice of this is hereby taken. "When there is a design need for market pressure to solve a problem and there are a finite number if identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp." *KSR*, 82 USPQ2d at 1397. It would have been obvious to try each of theses implementations to take advantage of the benefits each technique offers in order to achieve the best performance.

As per claim 15: Afsar discloses all the limitations of claim 1 as discussed above. Afsar further discloses wherein said manipulator manipulates includes: a data judger that, in the case where said judger judges that the condition is satisfied, judges whether data corresponding to the address generated by said address generator is stored in the cache **(see column 8, lines 18-33)** and a selector that selects a line within the cache in which data is stored, in the case where it is judged that the data is stored **(see column 8, lines 18-30)**. Afsar fails to disclose an invalidator that invalidates the selected line. Whenever data that is held in a cache memory is updated somewhere else in the system, and the data line no longer contains valid data, the data line is invalidated, this is a common memory coherence technique to make sure only the current version of data is used, and Official Notice is hereby taken. It would have been obvious to a person having ordinary skill in the art to which said subject matter pertains to have a data invalidation unit in the system of Afsar to invalidate data in the cache when that data line is updated somewhere else and is no longer valid to make sure any programs that use that data only use the current version of the data.

As per claim 16: Afsar discloses all the limitations of claim 1 as discussed above. Afsar further discloses wherein said manipulator includes: a data judger that judges, in a case where said judger judges that the condition is satisfied, whether data corresponding to the address generated by said address generator is stored in the cache **(see column 8, lines 18-33)** and a selector that selects a line, within the cache, in which data is stored, the case where it is judged that the data is stored **(see column 8, lines 18-30)**. Afsar fails to disclose an updater that changes an access order of lines based on selection by said selector. Cache replacement techniques are well known in the art. Selecting a replacement line based on the order in which the data is accessed allows older data to be replaced before newer data is, thereby keeping more useful data in the cache, and Official Notice is hereby taken. It would have been obvious to a person having ordinary skill in the art to which said subject matter pertains to have used an access order replacement technique in the system of Afsar to keep the most relevant data in the cache and to replace the older data.

As per claim 17: Afsar discloses all the limitations of claim 12 as discussed above. Afsar further discloses wherein said condition generator generates a memory address as the condition **(see column 6, lines 1-21)**. Afsar fails to disclose wherein said manipulator further includes an adjuster that adjusts, in a case where the memory address generated by said condition generator indicates a point midway through a line, to generate an address by adjusting so that one of a starting point of the line, a starting point of a next line, and a starting point of an immediately preceding line is indicated. When data is stored in an cache, the address of the line is maintained in a cache tag. Only a subset of the address is needed since each cache line stores a specific amount of data, the least significant bits are not needed to identify the line. This means that the addresses must occur in specific intervals depending on the design of the system (e.g. 16K, 32K, 64K, etc.). Official Notice of this is hereby taken. Therefore, if an address is generated that falls between these boundaries, the address will have to be adjusted to coincide with the size of the address that is maintained in the cache tag. It would have been obvious to a person having ordinary skill in the art to which said subject matter pertains to have modified the system disclosed by Afsar to modify any predicted addresses to coincide with the cache line boundaries and to adhere to the requirements of the cache tag array.

Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Afsar (**U.S. Patent #6,401,193**) in view of Yamada **(JP 07-084879)** and well known practices in the art.

As per claim 13: Afsar discloses all the limitations of claim 1 as discussed above. Afsar further discloses wherein said manipulator includes: a data judger that judges, in a case where said judger judges that the condition is satisfied, whether data corresponding to the address generated by said address generator is stored in the cache **(see column 8, lines 18-33)** and a selector that selects a line within the cache in a case where it is judged that the data is not stored **(see column 8, lines 18-30)**. Afsar fails to disclose an updater that performs a write back from the selected line when the selected line is valid and dirty and a registerer that registers the generated address as a tag, to the selected line, without transferring data from a memory to the selected line. There are two methods for ensuring that data that is written to a cache memory is updated in main memory. A write though method which involves updating data in main memory whenever the data in the cache line is updated and a write-back method that involves updating data in the main memory whenever the data is evicted from the cache memory. Both techniques have their advantages. A write thought method puts more traffic on the memory bus as data is constantly being sent to the main memory to update any data lines that have been written to, but allows faster eviction as the data can just be cast out without having to first wait for the data in main memory to be updated. A write back technique reduces the amount of traffic on the memory bus, but requires a wait time when evicting the data since the data in the main memory must first be updated so that any changes made are not lost. Official Notice of this is hereby taken. “When there is a design need for market pressure to solve a problem and there are a finite number if identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp." *KSR*, 82 USPQ2d at 1397. It would have been obvious to try each of theses implementations to take advantage of the benefits each technique offers in order to achieve the best performance. The combination still fails to disclose a registration unit operable to register the generated address as a tag, to the selected line, without transferring data from a memory to the selected line. Yamada discloses a technique of loading an address into a cache memory without transferring the data associated with the address from main memory to the cache because the data is just going to be over written **(see [0035]-[0036])**. Since the data is just going to be over written, there is no need to load the data from main memory. This will subsequently reduce the load on the bus **(see [0036])**. It would have been obvious to a person having ordinary skill in the art to which said subject matter pertains to have modified the combination to allow registering of a cache line without transferring the data from main memory when the data is just going to be over written anyway, thereby reducing the load on the bus, as taught by Yamada.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Edward J. Dudek whose telephone number is 571-270-1030. The examiner can normally be reached on Mon thru Thur 7:30-5:00pm Sec. Fri 7:30-4 pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Matt Kim/
Supervisory Patent Examiner, Art
Unit 2186

/E. J. D./
Examiner, Art Unit 2186
May 22, 2009